REMARKS

Claim 1-39 are pending. Claims 1, 12, 22 and 32 are independent.

Applicant presents for the examiner's reference amendments to claim 20 and 32 that were made in applicant's Amendment in Reply to Action of December 28, 2006, but with respect to which, the claims' status indicators erroneously indicated "Original" instead of "Currently amended." The amendments to claims 20 and 32 corrected typographical errors. If these amendments were not previously entered, applicant requests that these amendments be entered.

Applicant amended claim 36 to add the word "further" preceding the wording "configured to", and to remove the wording "one of a result of the task and." Applicant similarly amended claims 37-39.

The examiner objected to specification on the ground the text includes some minor typographical errors. Applicant amended the specification to correct the errors cited. Applicant thanks the examiner for bringing these errors to the applicant's attention.

The examiner rejected claims 1-3, 12-14, 22-24 and 36-38 under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,068,821 to Sexton. Additionally, the examiner rejected claims 4-10, 15-21 and 25-31 under 35 U.S.C. §103(a) as being unpatentable over Sexton in view of the reference "The Next Generation of Intel IXP Network Processors" by Adiletta et al. Further, the examiner rejected claim 11 under 35 U.S.C. §103(a) as being unpatentable over Sexton in view of U.S. Patent Publication No. 2002/0143998 to Rajagopal et al., rejected claims 32 and 39 under 35 U.S.C. §103(a) as being unpatentable over Sexton in view of U.S. Patent Publication No. 2002/0009079 to Jungck et al, and rejected claims 33-35 under 35 U.S.C. §103(a) as being unpatentable over Sexton in view of Jungck and further in view of Adiletta.

Specifically, with respect to independent claim 1, the examiner stated:

> 12. As per claim 1, Sexton teaches a method of co-processing, comprising:
>
> connecting an interface of a first processor (element 110) to an interface of a second processor (element 120) using a bus, the interface of the second processor being configurable to place the second processor in a slave processing mode or a master processing mode (see e.g. Fig. 2. Also, see e.g. col. 5, lines 47-67 to col. 6, lines 1-64); and

> sending a task from the first processor to the second processor through the bus, the task comprises an instruction that places the second processor in a slave processing mode (see e.g. col. 5, lines 47-67, wherein the start command is the instruction). (Final Action, paragraph 9, page 3)

Additionally, responding to applicant's arguments presented in the previous Amendment in Reply, the examiner stated:

> The applicant admits "Sexton bit processor places itself in slave or master mode, for example, bit processor 120 places itself in master mode in response to a start command written by function block processor 110". The applicant claim recites the same limitation as indicated by Sexton; thus Sexton teaches the limitation of the claim. For example, claim 1 recites "**sending a task from the first processor to the second processor through the bus, the task comprises an instruction that places the second processor in a slave processing mode.**" Mapping the teachings of Sexton into the applicant's claim is as follows: "**sending a task from the first processor** (Sexton, Start command written by function block processor 110) **to the second processor through the bus, the task comprises an instruction that places the second processor in a slave processing mode** (Sexton, bit processor 120 places itself in slave mode in response to the start command (task), therefore the start command (task) places the second processor in a slave processing mode)."
>
> As for the applicant arguments with respect to the Interface "Indeed, Sexton does not even disclose interfaces used in conjunction with either of the processors" The examiner disagrees, the input/output ports of the processors (Sexton, Fig. 2, element 110 and element 120) are considered equivalent to interfaces. See also Foldoc.org, online computing dictionary (term: interface). Therefore, as the start command reaches the Input port of the second processor (Fig. 2, element 120), the input port (Interface) sends the start command to the second processor (Fig. 2, element 120) to places it in either slave or master mode. (Final Action, page 13)

Applicant disagrees with the examiner's contentions.

Applicant's independent claim 1 recites "[a] method of co-processing, comprising: connecting an interface of a first processor to an interface of a second processor using a bus, the interface of the second processor being configurable to place the second processor in a slave processing mode or a master processing mode."

In contrast, Sexton does not describe these features. Rather, Sexton describes programmable logic controller that includes a function block processor 110 for processing

function block instructions and a bit processor 120 for processing Boolean instructions (FIG. 2, Abstract, and col. 2, lines 36-59). Both the block processor 110 and the bit processor 120 are incorporated into the same single controller module (see FIG. 2, and col. 4, lines 38-42.) Sexton explains:

> **The operation of bit processor 120 in master mode is now discussed in more detail through reference to the operational flow chart of the master mode shown in FIG. 3. To initiate the master mode, function block processor 110 writes a start command to bit processor 120 as per block 200. A master bit is then written to master/slave mode register 121 as per block 205. Once this starting operation is performed, bit processor 120 isolates itself from function block processor 110 and asserts a WAIT signal on the WAIT line as per block 210. In response to the WAIT signal, function block processor 110 then executes an instruction which causes the function block processor to wait until bit processor 120 removes or de-asserts the WAIT signal. That is, function block processor 110 is permitted to respond to interrupts, but is forced to return to waiting once servicing of the interrupts is complete.**
>
> ...
>
> **However, if at descision [*sic*] block 220 a determination is made that the current command is a function block command with zero power flow or a function block command of the type having an OPCODE which must be processed by function block processor, then flow continues to block 225 at which a slave bit is written to mode register 121. This action returns bit processor 120 back to the slave mode. Bit processor 110 also de-asserts the WAIT signal to return control to function block processor 110 as per block 230. (Emphasis added, col. 6, lines 15-52)**

Thus, it is the bit processor 120 that places itself in master mode or in slave mode. Sexton does not describe that an interface of the bit processor 120 or of function block 110 causes the bit processor 120 to be placed in master or slave mode. Indeed, because the function block processor 110 and bit processor 120 are included in the same physical module (namely, the logic controller) and were thus designed to work in tandem, there is no need whatsoever for an interface to facilitate inter-operability between processors not specifically designed to work in tandem and configured to place a processor in master or slave mode. Accordingly, Sexton fails to disclose or suggest at least the feature of "connecting an interface of a first processor to an interface of a second processor using a bus, the interface of the second processor being configurable to place the second processor in a slave processing mode or a master processing

mode," as required by applicant's independent claim 1. Applicant's independent claim 1, and the claims that depend from it are therefore patentable over the cited art.

Applicant's independent claims 12 and 22 recite "a first processor having an interface connected to an interface of a second processor using a bus, the interface of the first processor being configurable to place the first processor in a slave processing mode or a master processing mode," or similar language. For reasons similar to those provided with respect to independent claim 1, at least this feature is not disclosed by the cited art. Accordingly, applicant's independent claims 12 and 22, and the claims that respectively depend from them, are therefore patentable over the cited art.

As noted, the examiner rejected independent claim 32 as being unpatentable over Sexton in view of Jungck.

Applicant's independent claim 32 recites "[a] network router, comprising: a network co-processing system, the network co-processing system comprising: a first processor having an interface; and a second processor having an interface connected to the interface of the first processor by a bus, the interface of the second processor being configurable to place the second processor in a slave processing mode or a master processing mode, an input line connecting the network co-processing system to a first network; and an output line connecting the network co-processing system to a second network."

For reasons similar to those provided with respect to independent claim 1, Sexton does not disclose or suggest at least the feature of "a second processor having an interface connected to the interface of the first processor by a bus, the interface of the second processor being configurable to place the second processor in a slave processing mode or a master processing mode, and the Jungck reference is not seen to cure these deficiencies in Sexton.

Because neither Sexton nor Jungck discloses or suggests, alone or in combination at least the feature of "a second processor having an interface connected to the interface of the first processor by a bus, the interface of the second processor being configurable to place the second processor in a slave processing mode or a master processing mode," applicant's independent claim 32, and the claims that depend from it, are therefore patentable over the cited art.

As noted, the examiner rejected claim 36 as being anticipated by Sexton. Specifically, the examiner stated:

> 16. As per claim 36, Sexton teaches invention as claimed above. Sexton further teaches wherein the interface of the second processor is configured to provide the task to the second processor when the second processor is in slave mode (see Sexton, e.g. col. 5, lines 47-67 to col. 6, lines 1-64, wherein input port (interface) of the second processor receives the start command, this start command will be sent from the input port (interface) to the second processor), and to send one of a result of the task and another task to the first processor when the second processor is in master mode (see Sexton, e.g. col. 5, lines 47-67 to col. 6, lines 1-64). (Final Action, page 5, paragraph 16)

Applicant disagrees.

Applicant's claim 36 recites "wherein the interface of the second processor is further configured ... to send another task to the first processor when the second processor is in master mode."

Sexton explains:

> **However, if at descision block 220 a determination is made that the current command is a function block command with zero power flow or a function block command of the type having an OPCODE which must be processed by function block processor, then flow continues to block 225 at which a slave bit is written to mode register 121. This action returns bit processor 120 back to the slave mode. Bit processor 110 also de-asserts the WAIT signal to return control to function block processor 110 as per block 230. The function block processor 110 then reads the instruction pointer from bit processor 120 and thus is provided a reference to its OPCODE in user program RAM 135 as per block 240. It is noted that the OPCODE of the current command has no meaning to function block processor 110. However, the information following the OPCODE does have meaning to function block processor 110. Bit processor 120 then increments or otherwise adjusts program counter 122 to point to the address in user program RAM 135 of this next OPCODE as per block 235 thus updating an instruction pointer. (Emphasis added, col. 6, lines 42-63)**

Thus, Sexton describes that the function processor 110 receives an instruction pointer, pointing to an instruction that the function processor is to execute. Sexton does not describe that the bit processor 120 sends an instruction, and certainly not a task, to the function processor 110. It follows that any interface of the bit processor (assuming, for argument's sake, that the bit processor has an interface) does not send instructions or tasks to the function processor 110.

Accordingly, Sexton fails to disclose or suggests at least the features of "wherein the interface of the second processor is further configured ... to send another task to the first processor when the second processor is in master mode," as required by applicant's claim 36. Claim 35 is therefore patentable over the cited art.

Claims 37-39 recite "wherein the interface of the second processor is further configured ... to send another task to the first processor when the second processor is in master mode," or similar language. For reasons similar to those provided with respect to claim 36, at least this feature is disclosed by the cited art. Claims 37-39 are therefore patentable over the cited art.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

No fees are believed due. Please apply any required fees to deposit account 06-1050, referencing the attorney docket number shown above.

Respectfully submitted,

Date: May 22, 2007

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21645037.doc